

INFORMATION RETRIEVAL SYSTEM PADA PENCARIAN DOKUMEN DIGITAL PADA SMARTPHONE MENGGUNAKAN ALGORITMA BOYER MOORE

Gilang Yulianta¹, Yudi Setiawan², Diyah Puspitaningrum³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu.
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(telp: 0736-341022; fax: 0736-341022)

¹gilangyulianta07@gmail.com

²ys.teknik@gmail.com

³diyahpuspitaningrum@gmail.com

Abstrak: Aplikasi ini akan mencari dokumen berdasarkan kata kunci yang dimasukkan pengguna pada *smartphone* Android menggunakan Algoritma *Boyer Moore*. Pada pengujian aplikasi ini pengujian relevansi dokumen dilakukan oleh 3 pengguna *smartphone* Android pada dokumen berekstensi .doc, .pdf, dan .txt. Menggunakan kata kunci dengan panjang kata kunci sebanyak 1-3 kata. Panjangnya kata kunci akan diproses per karakter oleh Algoritma *Boyer Moore* sehingga yang mempengaruhi lamanya waktu eksekusi adalah banyaknya total karakter pada keseluruhan kata kunci. Metode ini adalah metode *Exact Match*. Hasil pengujian menunjukkan sistem pencarian file dengan metode *Boyer Moore* adalah jauh lebih baik di bandingkan dengan *File Manager* yang ada pada *smartphone* Android dengan *Precision @3* tertinggi adalah 0.9 serta MRR tertinggi mencapai 1. Hasil eksperimen waktu pencarian terburuk adalah selama 4 menit 9 detik yaitu pada pencarian dokumen .pdf. Panjangnya jalur pencarian sangat mempengaruhi, dimana semakin banyak *folder* dan *file* yang terlibat maka semakin lama waktu eksekusi yang dibutuhkan.

Kata Kunci: *Android, Boyer Moore, Relevansi Dokumen, Information Retrieval*

Abstract: The application searched for documents based on keywords entered by the user on the Android smartphone use Boyer Moore Algorithm. In this application testing, the relevance of document 3 Android smartphone had done by users on the document with extension .doc, .pdf, and .txt. Use the keyword long as 1-3 keywords. The length of keyword will be processed character by Boyer Moore algorithm thus affecting the length of execution time is the total number of characters in the entire keyword. This method is Exact Match. The test results a file search system with Boyer Moore is much better in comparison with existing File Manager on Android smartphones with the highest Precision @ 3 is 0.9 and the highest MRR reaches 1. The experimental results were worst-case search time is 4 minutes 9 seconds is on.pdf document search. The length of search path influence, where a growing number of folders and files involved the longer execution time required.

Keywords: Android, Boyer moore, Document Relevance, Information Retrieval

I. PENDAHULUAN

Sistem operasi Android mendukung pembacaan dan pencarian serta menampilkan pencarian suatu dokumen *digital*. Sistem operasi Android juga mempunyai fitur pencarian dokumen *digital* seperti aplikasi *File Manager*. Sehingga pengguna memiliki kemudahan dalam mencari dokumen *digital* yang telah di unduh atau disimpan pada memori *smartphone*.

Algoritma *Boyer Moore* merupakan algoritma pencocokan *String* yang tepat dan terkenal algoritma pencocokan yang digunakan dalam pola tunggal dan menganggap sangat cepat dalam kinerjanya. Dengan menggunakan algoritma ini, secara rata-rata proses pencarian akan menjadi lebih cepat jika dibandingkan dengan algoritma lainnya. Alasannya adalah algoritma ini melakukan pencocokan dari kanan (posisi terakhir *pattern* yang di cari) [1].

Oleh karena itu pada penelitian ini dirancang sebuah aplikasi yang melakukan pencarian dengan algoritma *Boyer Moore* untuk melakukan pencocokan *string* dari judul dokumen dan isi dokumen. Tujuan dari penelitian ini adalah meningkatkan relevansi dokumen agar sesuai dengan kebutuhan informasi dari pengguna.

II. LANDASAN TEORI

A. Metrik Relevansi

Mean Reciprocal Rank (MRR) adalah posisi hasil relevan pencarian yang menurut pengguna sudah tepat. Rumus MRR dapat dilihat pada persamaan 2.1 sebagai berikut [2] :

$$MRR = \frac{1}{r(q)} \quad (2.1)$$

$r(q)$: posisi dari dokumen yang relevan.

Precision adalah nilai ketepatan dalam rasio tertentu. Rumus *precision* dapat dilihat pada persamaan 2.2 sebagai berikut [3] :

$$Precision = \frac{|[a] \cap [b]|}{|[b]|} \quad (2.2)$$

a : Jumlah dokumen yang relevan

b : Jumlah dokumen dalam rasio tertentu (P@3, P@5, P@10, dan P@15)

B. Algoritma Boyer Moore

Algoritma *Boyer-Moore* adalah salah satu algoritma pencarian string, dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian *string* yang ditemukan sebelumnya, Algoritma *Boyer-Moore* mulai mencocokkan karakter dari sebelah kanan *pattern*. Ide dibalik algoritma ini adalah bahwa dengan memulai pencocokkan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat [4].

Pada fase *preprocessing* terdapat dua buah fungsi untuk menggeser pola kearah kanan. Kedua fungsi ini disebut disebut *good-suffix-shift* dan *bad character-shift* adalah fungsi yang memiliki aturan dan cara perhitungan yang akan menghasilkan nilai untuk dijadikan acuan pergeseran *pattern* pada teks. Untuk hasil perhitungan *bad-character-shift* akan menghasilkan nilai *Occurrence Heuristic* (OH) pada tabel BmBc. Sedangkan untuk hasil perhitungan *good-suffix shift* akan menghasilkan nilai *Match Heuristic* (MH) pada tabel BmGs.

Pada sebuah pencarian dokumen, pengguna android ingin mencari dokumen yang dibaca. Pengguna mencari teks yang akan dicari adalah “analisis sistem informasi”, dan kata kunci yang akan dicari adalah “sistem informasi”. Pengguna ingin mencari dokumen secara tepat, dan memilih algoritma untuk pencarian dari teks tersebut.

Penyelesaian dari contoh kasus di atas jika menggunakan algoritma *Boyer Moore* adalah

sebagai berikut:

Contoh 1 :

Teks : *analisis sistem informasi*

Kata Kunci : *sistem informasi*

1) Cara membuat Tabel OH (*Occurrence Heuristic*)

- Lakukan pencacahan mulai dari posisi terakhir *string* sampai ke posisi awal, dimulai dengan nilai 1, catat karakter yang sudah ditemukan (dalam contoh ini karakter "I"). Mundur ke posisi sebelumnya, nilai pencacah ditambah 1, jika karakter pada posisi ini belum pernah ditemukan, maka nilai pergeserannya adalah sama dengan nilai pencacah. (dalam contoh ini, karakter "S" belum pernah ditemukan sehingga nilai pergeserannya adalah sebesar nilai pencacah yaitu 2).
- Mundur ke posisi sebelumnya, karakter "A" nilai pergeserannya 3 dan begitu juga untuk sebelum karakter "A" maka nilai pergeserannya adalah 4 sampai seterusnya.
- Karakter "I" sudah pernah ditemukan sebelumnya sehingga nilai pergeserannya sama dengan nilai pergeseran karakter "I" yang sudah ditemukan paling awal yaitu 0. Begitu seterusnya sampai posisi awal *string*.

Tabel 1 Tabel OH Kata Kunci "Sistem Informasi"

Posisi	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
String	S	I	S	T	E	M		I	N	F	O	R	M	A	S	I
OH	1	0	1	12	11	3	9	0	7	6	5	4	3	2	1	0

Nilai MH (*Match Heuristic*) didapat dari langkah-langkah sebagai berikut :

- Jika karakter pada posisi 16 bukan "I" maka geser 1 posisi, berlaku untuk semua *string* yang dicari.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
S	I	S	T	E	M		I	N	F	O	R	M	A	S	I									
S	I	S	T	E	M		I	N	F	O	R	M	A	S	I									

- Jika karakter "I" sudah cocok, tetapi karakter

sebelum "I" bukan "S" maka geser sebanyak 16 posisi, sehingga posisi pattern melewati teks karena sudah pasti "SISTEM INFORMASI" bukan "SISTEM INFORMASI"

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
S	I	S	T	E	M		I	N	F	O	R	M	A	S	I									
S	I	S	T	E	M		I	N	F	O	R	M	A	S	I									

- Jika karakter "SI" sudah cocok, tetapi karakter sebelum "SI" bukan "A" maka geser sebanyak 14 posisi, sehingga posisi *string* berada / bersesuaian dengan akhiran "SI" yang sudah ditemukan sebelumnya. karena bisa saja akhiran "SI" yang sudah ditemukan sebelumnya merupakan awalan dari pattern "SISTEM INFORMASI" yang berikutnya. selanjutnya sama, pergeseran paling mungkin dan aman dalam Tabel *Match Heuristic* adalah pergeseran sebanyak 14 posisi.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
S	I	S	T	E	M		I	N	F	O	R	M	A	S	I									
S	I	S	T	E	M		I	N	F	O	R	M	A	S	I									

Dari uraian diatas didapatkan hasil OH dan MH dari kata kunci. Tabel 2 merupakan uraian OH dan MH kata kunci.

Tabel 2 Nilai OH Dan MH dari Kata Kunci "sistem informasi"

Posisi	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
String	S	I	S	T	E	M		I	N	F	O	R	M	A	S	I
OH	1	0	1	12	11	3	9	0	7	6	5	4	3	2	1	0
MH	14	14	14	14	14	14	14	14	14	14	14	14	14	14	16	1

Tahap-tahap pencocokan kata kunci terhadap teks adalah sebagai berikut.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	N	A	L	I	S	I	S	I	S	T	E	M		I	N	F	O	R	M	A	S	I		
S	I	S	T	E	M		I	N	F	O	R	M	A	S	I									

Karakter *spasi* tidak cocok dengan karakter "I". Nilai OH karakter *spasi* nilai pergeserannya adalah 9. Nilai MH ketidakcocokan pada posisi 16 (karakter "I") nilai pergeserannya = 1. Sehingga geser sebesar 9 posisi (nilai maksimal dari kedua tabel pergeseran). Dari pergeseran sebelumnya maka kata kunci yang dicari adalah cocok dengan salah satu kata pada teks.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	N	A	L	I	S	I	S	I	S	T	E	M		I	N	F	O	R	M	A	S	I		
S	I	S	T	E	M		I	N	F	O	R	M	A	S	I									

Contoh 2 :

Teks : ANALISIS SISTEM INFORMASI

Kata Kunci : SISTEMINFORMASI

Kata kunci diatas didapatkan nilai OH dan Nilai MH sebagai berikut :

Tabel 2 Nilai OH Dan MH dari Kata Kunci “sisteminformasi”

Posisi	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
String	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1
OH	S	I	S	T	E	M	I	N	F	O	R	M	A	S	I	S
MH	1	0	1	1	1	0	3	0	7	6	5	4	3	2	1	0

Tahap-tahap pencocokan kata kunci terhadap teks adalah sebagai berikut.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	N	A	L	I	S	I	S	S	I	S	T	E	M	I	N	F	O	R	M	A	S	I		
S	I	S	T	E	M	I	N	F	O	R	M	A	S	I										

Karakter “M” tidak cocok dengan karakter “T”. Nilai OH karakter “M” nilai pergeserannya adalah 3. Nilai MH ketidakcocokan pada posisi 15 (karakter “T”) nilai pergeserannya = 1. Sehingga geser sebesar 3 posisi (nilai maksimal dari kedua tabel pergeseran).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	N	A	L	I	S	I	S	S	I	S	T	E	M	I	N	F	O	R	M	A	S	I		

Karakter “N” tidak cocok dengan karakter “T”. Nilai OH karakter “N” nilai pergeserannya adalah 7. Nilai MH ketidakcocokan pada Posisi 18(karakter “T”) nilai pergeserannya = 1. Sehingga geser sebesar 7 posisi (nilai maksimal dari kedua tabel pergeseran).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	N	A	L	I	S	I	S	S	I	S	T	E	M	I	N	F	O	R	M	A	S	I		

Karakter “INFORMASI” cocok dengan kata kunci karakter “INFORMASI”. Selanjutnya mencocokkan karakter sebelum kata “INFORMASI” yaitu karakter “M”. karakter *spasi* tidak cocok dengan karakter “M”. Nilai OH karakter “M” adalah 3 dan nilai MH posisi 16 (karakter “M”) nilai pergeserannya adalah 14. Nilai maksimal dari kedua nilai tersebut adalah 14 maka pergeseran yang dilakukan adalah sebanyak 14 karakter. Namun pergeseran ini tidak memungkinkan karena panjang pergeseran melebihi panjang karakter teks sehingga kata

kunci yang dicari tidak ditemukan. Dua contoh diatas dapat disimpulkan bahwa kinerja Algoritma *Boyer Moore* sangat sensitif sekali dengan karakter-karakter yang digunakan.

III. METODOLOGI PENELITIAN

A. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini yaitu dokumentasi dilakukan dengan cara mencari data atau dokumen *digital* dari Android yang berasal dari internet seperti *e-mail* dan dokumen yang disimpan berbentuk dokumen dengan format .doc, .txt dan .pdf.

B. Teknik Pra-pemrosesan

Pada penelitian dengan *Boyer Moore* ini dilakukan pencarian pada judul dan isi *file*. Kemudian tidak diperlukan pra-pemrosesan berupa pemecahan perkata karena *Boyer Moore* melakukan pembacaan karakter-karakter yang ada pada kata kunci.

C. Metode Pengujian

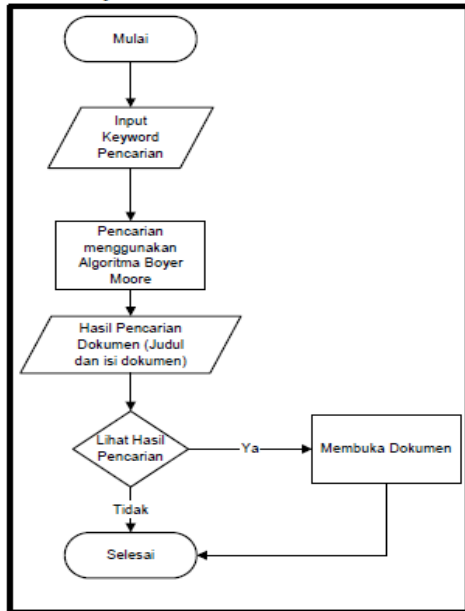
Proses pengujian yang dilakukan pada aplikasi yang dibuat menggunakan metode pengujian yaitu *black box testing*. Pada pengujian ini dilakukan dengan dengan tujuan untuk mengetahui nilai *confidence* dari relevansi, *Mean Reciprocal Rank* (MRR), dan *Precision*. Proses penilaian dilakukan oleh 3 orang pengguna (*Human Judgement*) dengan menentukan kerelevanan hasil pencarian, pengguna menilai dari hasil pencarian dengan obyektif. Selain itu kata kunci yang disiapkan sebanyak 20 kata kunci yang berbeda dari setiap ekstensi dokumen. Jumlah dokumen dalam pengujian ini adalah 100 dokumen setiap ekstensinya.

IV. ANALISIS DAN PERANCANGAN

Pada analisis cara kerja sistem akan dilakukan dengan menganalisa tentang bagaimana cara perangkat lunak dari sistem tersebut dapat berinteraksi dengan pengguna, mulai dari

memasukkan teks judul dokumen sampai mengeluarkan keluaran berupa susunan atau *query* dari hasil masukan teks judul sebelumnya. Tahapan perancangan ditunjukkan pada Gambar 1 berikut :

A. Cara Kerja Sistem



Gambar 1 Diagram Alir Aplikasi Pencarian Dokumen

Berdasarkan Gambar Keterangan diagram alir :

1. Dalam aplikasi ini pengguna diminta untuk memasukkan *keyword* pencarian berdasarkan apa yang akan dicari. Jika sebelumnya pengguna sudah melakukan pencarian maka pengguna akan melihat *recent searching*.
2. Setelah semua data yang diperlukan telah dimasukkan maka sistem akan melakukan pencarian yang dalam hal ini yang sangat berperan adalah Algoritma *Boyer Moore* dan akan menampilkan hasil pencarian berdasarkan *keyword* yang dimasukkan.
3. Jika pengguna ingin melihat *file*, pengguna harus memilih salah satu nama *file* dari daftar hasil pencarian, didukung dengan aplikasi tambahan yang berasal dari Android itu sendiri seperti *OpenOffice* atau aplikasi *Office* lainnya. Apabila pengguna tidak ingin membuka *file* maka kerja sistem selesai.

V. PEMBAHASAN

A. Pengujian Kerelevanan Hasil Pencarian Berdasarkan Jenis Ekstensi

Pada pengujian ini memiliki tujuan untuk mengetahui tingkat kerelevanan dan ketepatan dari hasil pencarian terhadap pengguna atau responden. Aplikasi ini menerapkan *First In First Out* (FIFO) dalam menampilkan hasil pencarian artinya ketika aplikasi sudah menemukan dokumen yang dicari maka dokumen tersebut langsung ditampilkan. Dokumen dikatakan ditemukan jika kata kunci ditemukan pada judul atau isi. Artinya jika kata kunci ditemukan pada judul maka tidak perlu memproses pada isi *file*. *Worst case* adalah ketika ditemukan pada penghujung isi *file* atau tidak ditemukan sama sekali.

Pengujian ini diukur dengan poin-poin penting yaitu dengan pengujian *Mean Reciprocal Rank* (MRR), kerelevanan *confidence* dari relevansi, dan *precision*. *Mean Reciprocal Rank* (MRR) adalah posisi hasil pencarian yang menurut pengguna sudah tepat contohnya seperti pengguna menemukan hasil pencarian yang menurutnya tepat berada di posisi pertama. Jadi MRR dari hasil pencarian adalah berada di posisi pertama

Precision adalah nilai ketepatan dalam rasio tertentu. Selain itu, dalam pencarian dokumen jumlah dokumen terdapat 100 dokumen dari masing-masing ekstensi dokumen. Kata Kunci dalam pengujian ini terdapat 20 kata kunci dari setiap ekstensi. Ekstensi dari dokumen adalah .doc, .pdf, dan .txt. Tabel A.1 pada Appendix menjelaskan bahwa waktu terbaik didapat pada kata kunci “Android” lebih cepat dibandingkan kata kunci “Pengolahan Citra Digital” ini karena dalam penerapannya Algoritma *Boyer Moore*

melakukan pencocokan secara per-karakter termasuk *spasi* sehingga banyaknya jumlah kata hanya dibaca *unigram*, artinya apabila jumlah kata yang dimiliki adalah sebanyak 3 kata jadi yang dibaca kata kunci secara keseluruhan tidak hanya 1 atau 2 kata saja.

Algoritma ini melakukan pencocokan per-karakter sehingga algoritma ini memerlukan waktu yang lebih lama dalam mencocokkan kata kunci “Pengolahan Citra Digital” yang memiliki jumlah karakter sebanyak 24 karakter lebih banyak dibandingkan dengan karakter “Android” yang memiliki karakter sebanyak 7 karakter.

Dari Tabel A.2 pada Appendix dapat dilihat bahwa $P@3$ kata kunci yang terbaik adalah kata kunci “Tujuan Penelitian”, “Proposal Skripsi”, dan “Tinjauan pustaka” karena hasil pencarian dari kata kunci tersebut mendapatkan nilai rata-rata $P@3$ adalah 1, Sedangkan kata kunci pada $P@3$ yang terburuk adalah “Pengolahan Citra Digital” karena hasil pencarian dari kata kunci tersebut mendapatkan nilai rata-rata $P@3$ adalah 0.6.

Berdasarkan Tabel A.1-A.3 pada Appendix sampai dengan *precision* $P@3$ pada sistem adalah cukup tinggi yaitu 0.8, 0.9, dan 0.8 masing-masing untuk .doc, .pdf, dan .txt. Dari Tabel A.3 pada Appenfix ada beberapa kata kunci yang memiliki rata-rata *precision* $P@3$ sama namun waktu eksekusi yang berbeda, contohnya kata kunci “Perangkat Keras” (15 karakter) yang memiliki nilai rata-rata $P@3=0.7$ dan memiliki waktu eksekusi 46.8 detik dibandingkan kata kunci “Sistem Informasi” (16 karakter) yang memiliki nilai $P@3=0.7$ dengan waktu eksekusi 40.6 detik, hal ini disebabkan karena pengaruh panjang karakter judul dan isi *file* letaknya dalam *folder* atau *subfolder*. Adapun rata-rata MRR total untuk Tabel A.1, Tabel A.2, Tabel A.3 adalah 0.9, 1, 0.9.

B. Pengujian Perbandingan dengan Aplikasi File Manager

Dari Tabel 4 dapat dilihat bahwa aplikasi yang menggunakan algoritma *Boyer Moore* memiliki hasil yang cukup baik karena hasil pencarian dokumen lebih banyak dibandingkan aplikasi *File Manager*. *File manager* tidak memiliki hasil yang banyak dan variatif karena sistem pencarian yang dilakukan oleh *File Manager* hanyalah mencari judul *file* saja, berbeda dengan pada *Boyer Moore* yang melakukan pencarian judul *file* dan isi dari *file*

Tabel 4 Perbandingan Jumlah dari Hasil Pencarian Berekstensi .doc

No.	Kata kunci	Jumlah Hasil Pencarian	
		Aplikasi Menggunakan Boyer Moore	Aplikasi File Manager
1.	Universitas Bengkulu	20 Dokumen	-
2.	Kuliah Kerja Nyata	18 Dokumen	-
3.	Program Kerja	17 Dokumen	1 Dokumen
4.	Pemrograman	20 Dokumen	2 Dokumen
5.	Internet	20 Dokumen	1 Dokumen
6.	Perancangan Sistem	18 Dokumen	-
7.	Website	18 Dokumen	-
8.	Analisis Kebutuhan	13 Dokumen	-
9.	Landasan Teori	14 Dokumen	-
10.	Manfaat Penelitian	10 Dokumen	-
11.	Tujuan Penelitian	12 Dokumen	-
12.	Metodologi Penelitian	10 Dokumen	-
13.	Proposal Skripsi	10 Dokumen	4 Dokumen
14.	Basis Data	10 Dokumen	-
15.	Android	10 Dokumen	-
16.	Metode Pengembangan Sistem	10 Dokumen	-
17.	Pengolahan Citra Digital	5 Dokumen	-
18.	Tinjauan Pustaka	10 Dokumen	-
19.	Tumbuhan Paku	8 Dokumen	-
20.	Sistematika Penulisan	7 Dokumen	-

C. Pengujian Terhadap Panjang Path Pencarian

Pengujian ini terdapat beberapa *path* pencarian pada masing masing jenis ekstensi dokumen. Pada pengujian ini 20 kata kunci dicari dengan 2 *path* yang berbeda. Tabel 5, Tabel 6, dan Tabel 7 merupakan perbandingan pengaruh 2 *path* pencarian terhadap waktu eksekusi.

Tabel 5 Perbandingan *Path* Pencarian Terhadap Waktu Eksekusi ekstensi .doc

No.	Katakunci	/storage/emulata/0/ DOC	/storage/sdcard0/Download /
1.	Universitas Bengkulu	57.3 detik	60.7 detik
2.	Kuliah Kerja Nyata	61.7 detik	74.3 detik
3.	Program Kerja	56.4 detik	63.3 detik
4.	Pemrograman	56.4 detik	62.3 detik
5.	Internet	60.3 detik	73.6 detik
6.	Perancangan Sistem	76.6 detik	84.2 detik
7.	Website	70.5 detik	87.2 detik
8.	Analisis Kebutuhan	98.7 detik	100.2 detik
9.	Landasan Teori	60.4 detik	72.3 detik
10.	Manfaat Penelitian	60.1 detik	73.6 detik
11.	Tujuan Penelitian	57.3 detik	64.8 detik
12.	Metodologi Penelitian	52.8 detik	63.9 detik
13.	Proposal Skripsi	64.6 detik	73.7 detik
14.	Basis Data	67.9 detik	72.5 detik
15.	Android	64.5 detik	71.3 detik
16.	Metode Pengembangan Sistem	64.2 detik	73.6 detik
17.	Pengolahan Citra Digital	106.6 detik	122.6 detik
18.	Tinjauan Pustaka	72.6 detik	84.7 detik
19.	Tumbuhan Paku	59.3 detik	58.3 detik
20.	Sistematika Penulisan	43.2 detik	50.7 detik
		65.57 detik	74.39 detik

Adapun jumlah file pada path /storage/emulate/0/ DOC = 100 *file* dan pada path /storage/sdcard0/Download / = 300 *file*.

Tabel 6 Perbandingan *Path* Pencarian Terhadap Waktu Eksekusi ekstensi .pdf

No.	Kata Kunci	/storage/emulate/0/ DOC	/storage/sdcard0/Download /
1.	Jurnal	230.6 detik	248.4 detik
2.	Sistem Pendukung Keputusan	273.3 detik	287.8 detik
3.	Game	298.7 detik	310.7 detik
4.	Augmented Reality	286.3 detik	298.5 detik
5.	Sistem Informasi	296.7 detik	305.8 detik
6.	Tumbuhan	275.2 detik	286.9 detik
7.	Pembelajaran	266.3 detik	274.9 detik
8.	Jaringan	219.2 detik	228.6 detik
9.	Obat	282.3 detik	296.8 detik
10.	Boyer Moore	235.8 detik	254.9 detik
11.	Rancang Bangun	222.9 detik	246.7 detik
12.	Biologi	276.8 detik	240.5 detik
13.	String Matching	272.9 detik	288.6 detik
14.	Aplikasi	257.2 detik	261.3 detik
15.	Android	217.9 detik	248.5 detik
16.	Penyakit	22.17 detik	225.7 detik
17.	Kesehatan	233.2 detik	230.7 detik
18.	Laporan	235.7 detik	243.6 detik
19.	Fuzzy	271.9 detik	287.9 detik
20.	Mobile	220.7 detik	236.4 detik
		244.8 detik	265.2 detik

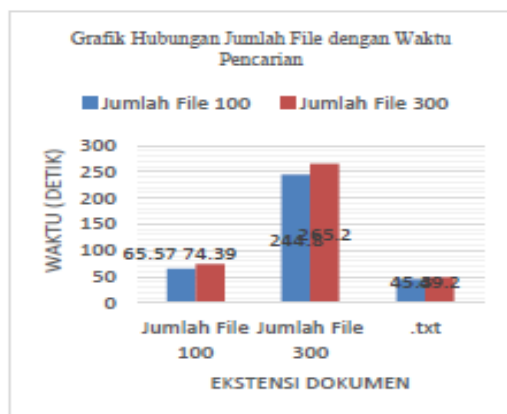
Adapun jumlah file pada path /storage/emulate/0/ PDF = 100 *file* dan pada path /storage/sdcard0/Download / = 300 *file*.

Tabel 7 Perbandingan *Path* Pencarian Terhadap Waktu Eksekusi ekstensi .txt

No.	Katakunci	/storage/emulate/0/ TXT	/storage/sdcard0/ Download /
1.	Jaringan Komputer	35.6 detik	36.5 detik
2.	Perangkat Keras	46.6 detik	37.2 detik
3.	Perangkat Lunak	38.1 detik	26.2 detik
4.	Protokol	37.4 detik	37.9 detik
5.	Rekayasa Perangkat Lunak	46.6 detik	45 detik

6.	sistem informasi	40.6 detik	37.2 detik
7.	sistem pakar	42.5 detik	35.4 detik
8.	TCP/IP	44.9 detik	44.3 detik
9.	Algoritma Enkripsi	55.4 detik	44.7 detik
10.	Algoritma Kriptografi	42.5 detik	43.9 detik
11.	Keamanan Jaringan	69.4 detik	72.3 detik
12.	Kunci Publik	49.9 detik	51.2 detik
13.	PHP	56.8 detik	58.3 detik
14.	Sistem Informasi Geografis	59.5 detik	61.3 detik
15.	Topologi Jaringan	55.8 detik	56.2 detik
16.	Fungsi Hash	66.5 detik	68.8 detik
17.	JAVA	54.5 detik	57.2 detik
18.	Kecerdasan Buatan	47.4 detik	49.8 detik
19.	Kriptografi Modern	44.7 detik	46.7 detik
20.	Model Waterfall	32.3 detik	34.5 detik
		45.3 detik	49.2 detik

Adapun jumlah file pada path /storage/emulate/0/PDF = 100 file dan pada path /storage/sdcard0/Download / = 300 file.



Gambar 2 Grafik Hubungan Jumlah File dengan Waktu pencarian

Dari Gambar 2 dapat dilihat bahwa panjang *Path* dan Jumlah *File* yang terlibat mempengaruhi waktu eksekusi pencarian. Jadi semakin banyak *file* yang terlibat maka semakin lama pula waktu eksekusi yang berjalan. Selain itu dokumen ekstensi .pdf lebih lama di bandingkan dengan dokumen ekstensi .doc dan .txt ini dikarenakan dalam proses pencarian dokumen berekstensi .pdf membutuhkan *library* tambahan untuk membantu algoritma *Boyer Moore* dalam proses pencarian. Dokumen ekstensi .txt tidak menggunakan *library* tambahan dalam proses pencarian.

VI. KESIMPULAN

- 1) Berdasarkan analisa perancangan sistem, implementasi dan pengujian sistem, maka dapat disimpulkan bahwa Algoritma *Boyer Moore* yang diimplementasikan pada *Information Retrieval* Pencarian dokumen digital pada *smartphone* memberikan hasil yang optimal.
- 2) Algoritma *Boyer Moore* dalam proses pencocokan jumlah karakter yang lebih banyak memerlukan waktu eksekusi yang lama dibandingkan dengan jumlah karakter yang lebih sedikit. Berdasarkan analisis yang dikerjakan bahwa MRR yang didapat pada dokumen .doc, .pdf, .txt adalah 0.9, 1, 0.9, sehingga tingkat dari MRR cukup tinggi. P@3 yang didapat pada dokumen .doc, .pdf, .txt adalah 0.8, 0.9, 0.8, (tingkat *precision* cukup tinggi). Hubungan antara hasil dari P@3 dan waktu eksekusi tidak ada keterkaitannya karena berbedanya cara mendapatkan nilainya masing-masing. Hal ini juga berlaku pada dokumen ekstensi .doc dan .txt
- 3) Pencarian dengan Algoritma *Boyer Moore* lebih efektif dibandingkan metode yang digunakan pada aplikasi *File Manager* yang tersedia di *smartphone*.
- 4) Banyaknya *path* pencarian juga mempengaruhi waktu eksekusi. Semakin banyak folder pada *path* pencarian dan semakin banyak *file* yang terlibat maka waktu eksekusi akan semakin lama dari pada *path* yang memiliki sedikit *folder* maupun *subfolder* yang terlibat.
- 5) Pada *Boyer Moore* panjang kata kunci sangat berpengaruh pada lamanya proses pencarian dimana baik 1, 2, atau 3 kata kunci akan diperlakukan sama seperti 1 kata dan diproses oleh *Boyer Moore* secara perkata.

- 6) Kelemahan dari algoritma *Boyer Moore* adalah ketika semua karakter memiliki kesamaan atau kecocokan dan hanya karakter terakhir atau karakter paling kiri yang berbeda maka pencarian ini akan memerlukan waktu yang sedikit lama.

VII. SARAN

Berdasarkan pada hasil penelitian, untuk pengembangan yang akan datang penulis menyarankan adalah alternatif untuk meningkatkan kecepatan dan kerelevanan pencarian yaitu dengan menggunakan *indexing*. Cara kerja *indexing* adalah dengan melakukan *scan* kata-kata penting (kata kunci) dalam dokumen kemudian dokumen tersebut disimpan dalam 1 *folder* dalam format *.idx atau format lainnya, sehingga apabila pengguna akan melakukan pencarian kembali aplikasi akan mengakses *folder indexing* tersebut.

REFERENSI

- [1] G. G. Leonaerde, 2014, "Penerapan Algoritma Boyer Moore pada Aplikasi Pengajuan Judul Skripsi Berbasis Web," *Jurnal Informasi dan Teknologi Ilmiah (INTI)*.
- [2] T. Y. Liu, 2009, *Learning to Rank for Information Retrieval*, Beijing: Microsoft Research Asia.
- [3] L. Robinson, 2012, "Implementasi Metode Generalized Vector Space Model Pada Aplikasi Information Retrieval untuk Pencarian Informasi Pada Kumpulan Dokumen Teknik Elektro Di UPT BPI LIPI," *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*.
- [4] M. . Y. Sholeh, 2010, "Implentasi Algoritma KMP dan Boyer-Moore dalam Aplikasi Search Engine Sederhana," *Jurnal ITB*.

APPENDIX

Tabel A.1 Ringkasan Hasil Pengujian 20 Kata Kunci dari Dokumen Berekstensi .doc

No	Kata Kunci	P@3			Rata-Rata P@3	Waktu Eksekusi (Detik)
		A	B	C		
1.	Universitas Bengkulu	2/3	3/3	2/3	0.7	57.3
2.	Kuliah Kerja Nyata	2/3	3/3	2/3	0.7	61.7
3.	Program Kerja	1/3	3/3	1/3	0.7	56.4
4.	Pemrograman	2/3	2/3	2/3	0.7	56.4
5.	Internet	3/3	2/3	0	0.7	60.3
6.	Perancangan Sistem	2/3	3/3	2/3	0.7	76.6
7.	Website	2/3	2/3	2/3	0.7	70.5
8.	Analisis Kebutuhan	2/3	2/3	2/3	0.7	98.7
9.	Landasan Teori	2/3	3/3	3/3	0.9	60.4
10.	Manfaat Penelitian	3/3	3/3	1/3	0.7	60.14
11.	Tujuan Penelitian	3/3	3/3	3/3	1	57.3
12.	Metodologi Penelitian	3/3	2/3	3/3	0.9	52.8
13.	Proposal Skripsi	3/3	3/3	3/3	1	64.6
14.	Basis Data	1/3	3/3	1/3	0.7	67.9
15.	Android	3/3	3/3	2/3	0.9	43.2
16.	Metode Pengembangan Sistem	3/3	3/3	2/3	0.9	64.2
17.	Tinjauan Pustaka	3/3	3/3	3/3	1	72.6
18.	Pengolahan Citra Digital	1/3	3/3	1/3	0.6	106.6
19.	Tumbuhan Paku	2/3	1/3	3/3	0.7	59.3
20.	Sistematika Penulisan	2/3	3/3	3/3	0.9	64.5
Rata-Rata Total		P@3 = 0.8				65.59

Tabel A.2 Ringkasan Hasil Pengujian 20 Kata Kunci dari Dokumen Berekstensi .pdf

No.	Kata Kunci	P@3			Rata-Rata P@3	Waktu Eksekusi (Detik)
		A	B	C		
1.	Jurnal	3/3	3/3	3/3	1	230.6
2.	Sistem Pendukung Keputusan	3/3	2/3	3/3	0.9	298.7
3.	Game	3/3	3/3	3/3	1	273.3
4.	Augmented Reality	3/3	3/3	3/3	1	286.3
5.	Sistem Informasi	3/3	3/3	2/3	0.9	296.7
6.	Tumbuhan	3/3	2/3	3/3	0.9	275.2
7.	Pembelajaran	3/3	3/3	3/3	1	266.3
8.	Jaringan	3/3	3/3	3/3	1	219.2
9.	Obat	2/3	2/3	3/3	0.8	282.3
10.	Boyer Moore	3/3	3/3	3/3	1	235.8
11.	Rancang Bangun	3/3	3/3	3/3	1	222.9
12.	Biologi	3/3	2/3	3/3	0.9	276.8
13.	String Matching	3/3	3/3	3/3	1	272.9
14.	Aplikasi	3/3	3/3	3/3	1	257.2
15.	Android	3/3	2/3	3/3	0.9	217.9
16.	Penyakit	3/3	2/3	3/3	0.9	221.7
17.	Kesehatan	3/3	3/3	3/3	1	233.2
18.	Laporan	3/3	3/3	3/3	1	235.7
19.	Fuzzy	3/3	3/3	3/3	1	271.9
20.	Mobile	3/3	2/3	3/3	0.9	220.7
Rata-Rata Total		P@3 = 0.9				243.6